

What Is Claimed Is:

1. A method of managing file-based data which is manipulated by at least one user via a file-based computerized editor, the file-based data including a plurality of file elements, the method comprising the steps of:

- (a) representing file-based data as a plurality of individual components, each individual component having a unique identity and identifier; and
- (b) storing said individual components in a store.

2. The method according to claim 1, wherein step (b) further comprises: storing all of said individual components corresponding to one file of said file-based data in a single store, wherein said single store contains no individual components of a different file of said file-based data.

3. The method according to claim 1, further comprising:

- (c) retrieving said components using a client comprising a private store, and a run-time agent, wherein said run-time agent looks up a store's server, connects with said store's server, requests said components from said server, and stores a version of said components in said private store.

4. The method according to claim 3, further comprising:

- (d) providing access to said retrieved components to external applications through said run-time agent.

009234-0004

5. The method according to claim 1, further comprising:

(c) recreating equivalent file-based data for use within an environment of a file-based computerized editor from said individual components in said store.

6. The method according to claim 5, wherein said recreated file-based data may be edited via a file-based editor in a plurality of successive sessions, and step (b) includes storing

(i) a latest version of the individual components, and

(ii) information to fully document changes made to each version of each individual component during each of said plurality of successive sessions, wherein said changes comprise additions, modifications and deletions.

7. The method according to claim 6, further comprising at least one of:

(d) using said information stored in step (b) (ii) to view successive versions of any individual changed component; and

(e) using the information stored in step (b) (ii) to view a plurality of changed components.

8. The method according to claim 1, wherein step (a) includes defining and storing a schema for said plurality of components, said schema being a set of classes that captures all of the information in said file-based data.

9. The method according to claim 8, further comprising the step of:

15. The method according to claim 1, wherein step (a) includes mapping at least some of said plurality of elements to respective single components.

16. The method according to claim 1, further comprising:

- (c) defining a tag for at least some of said file elements during interaction with said computerized editor; and
- (d) storing and saving a mapping between said tag for each tagged file element and its component identifier.

17. A data management structure for engineering design data comprising a plurality of components, each component having (i) a unique identifier, (ii) a set of fields, each field having a data type and a data value, and (iii) a program which interprets and modifies said fields, at least some of the components representing respective elements in a file-based computerized editing system.

18. The data management structure according to claim 17, wherein at least some of the components further have at least one of:

- (iv) a list of other dependent components; and
- (v) an access control value.

19. The data management structure according to claim 17, wherein a user manipulates said elements in a plurality of editing sessions using at least one of a file-based editing program and a component-based computerized modeling system, said

22. The apparatus according to claim 21, wherein at least some of said individual components further have (iv) a list of other dependent components, said memory further storing such lists.

23. The apparatus according to claim 21, wherein at least some of said individual components further have (iv) an access control value, said memory further storing such values.

24. The apparatus according to claim 21, wherein each element is represented by a component.

25. An apparatus for translating a plurality of individual components which are stored in a store into file-based data, each individual component having (i) a unique identifier, (ii) a set of fields, each field having a data type and a data value, and (iii) a program which interprets and modifies said fields, the apparatus comprising:

(a) a translation means for representing the components as file-based data, the file-based data including a plurality of file elements; and

(b) a display for displaying said file-based data in a file-based computerized editor to allow for manipulation by a user in said file-based format.

26. The apparatus according to claim 25, wherein at least some of the individual components further have at least one of:

(iv) a list of other dependent components; and

(v) an access control value.

27. The apparatus according to claim 25, wherein each component represents an element.

28. A client-server system for multi-user management of engineering data, the system comprising:

(a) a server including a store for storing a current version of a plurality of components, said components representing elements of at least one engineering project, each component having (i) a unique identifier, (ii) a set of fields, each field having a data type and a data value, and (iii) a program which interprets and modifies said fields; and

(b) a plurality of client computers, each client computer being bidirectionally connected to said server for receiving said current version of a plurality of components representing a subset of elements of an engineering project, and for sending locally edited versions of said components back to the server, each client computer including a component memory for storing a local version of said components.

29. The system according to claim 28, wherein components in said component memory of each client computer may be locally edited in successive editing sessions, and said component memory stores (i) a latest version of the plurality of individual components, and (ii) information to fully document changes made to each version of each individual component during said local editing.

30. The system according to claim 28, wherein each client computer includes means for allowing locally edited versions of said components to replace said latest current version of the components in said server's store only if no unresolved component conflicts exist between the two versions.

31. The system according to claim 28, wherein each client computer includes means for updating the latest locally edited version of the components to reflect any component changes made in the store since the client computer originally received the components from a server's store as a result of component changes made by other client computers in the time period subsequent to the original receipt, the means for updating including:

- (i) means for requesting and receiving said latest version of the components from said store's server,
- (ii) means for locally detecting resolvable and unresolvable component conflicts during said updating on a per component basis between said latest locally edited version of said components and said latest version of said components from said server's store; and
- (iii) means for replacing locally edited versions of said components, which have no conflict or no unresolvable conflict with a latest version of said components from said server's store.

32. The system according to claim 28 wherein said server stores a plurality of engineering projects and each client computer is adapted to receive, send and store said plurality of engineering projects.

33. The system according to claim 28, wherein said server comprises:
a plurality of stores for storing a current version of a plurality of components, said components representing elements of at least one engineering project; wherein each store stores the components corresponding to one design file; and wherein each component has (i) a unique identifier, (ii) a set of fields, each field having a data type and a data value, and (iii) a program which interprets and modifies said fields.

34. The client-server system of claim 33, wherein said server comprises a globally unique ID, a port number and an Internet address; and wherein said client computer further comprises a run-time agent for looking up a store's server with said server's Internet address and port number, connecting to said store's server, requesting said store's stored components and storing local versions of said components in said component memory.

35. The client-server system of claim 34, further comprising:
(a) a plurality of servers including a plurality of stores for storing a current version of a plurality of components, said components representing elements of at least one engineering project, each component having (i) a unique identifier, (ii) a set of fields,

each field having a data type and a data value, and (iii) a program which interprets and modifies said fields; and

(b) a plurality of client computers, each client computer being bidirectionally connected to said plurality of servers for receiving the current version of said plurality of components representing a subset of elements of an engineering project, and for sending locally edited versions of said components back to said plurality of servers, each client computer including a component memory for storing a local version of said components.

36. The client-server of claim 35, wherein said client computer is capable of receiving requests for said engineering data from other applications.

37. A computer program product embodied on a computer readable medium, said computer program product comprising program logic wherein the computer program product comprises:

translation program code means for representing file-based data as a plurality of individual components, each individual component having a unique identity and identifier;

storing program code means for storing said individual components in a store;

retrieval program code means for retrieving said components using a client comprising a private store, and a run-time agent, wherein said run-time agent looks up a store's server, connects with said store's server, requests said components from said server and stores a version of said components in said private store;

